

# Gestión de software centralizada con Ansible en un centro docente

Máster Universitario en Profesorado de Educación  
Secundaria Obligatoria y Bachillerato, Formación  
Profesional y Enseñanza de Idiomas

**Facultad de Educación**

Trabajo Fin de Máster

Autor:

ROBERTO ANGUITA MARTÍN

Tutor:

SONIA VÁZQUEZ PÉREZ

Junio 2021

## ÍNDICE

<b>1. INTRODUCCIÓN Y JUSTIFICACIÓN.....</b>	<b>5</b>
<b>2. DESARROLLO.....</b>	<b>7</b>
2.1. Contextualización .....	7
2.2. ¿Qué es Ansible? .....	10
2.3. Instalación del servidor .....	11
2.4. Instalación del cliente.....	13
2.5. Configuración del inventario .....	15
2.6. Configuración del playbook.....	15
<b>3. INFRAESTRUCTURA.....</b>	<b>18</b>
<b>4. POSIBLES IMPEDIMENTOS.....</b>	<b>19</b>
<b>5. CASUÍSTICAS .....</b>	<b>20</b>
<b>6. CONCLUSIONES / IMPLICACIONES.....</b>	<b>21</b>
<b>7. PROPUESTA DE MEJORA PARA LA IMPLEMENTACIÓN .....</b>	<b>22</b>
<b>8. BIBLIOGRAFÍA.....</b>	<b>22</b>
<b>9. ANEXOS.....</b>	<b>23</b>
Anexo 1: Ficheros YAML.....	23
Anexo 2: Módulos de Ansible .....	24
Gestor de paquetes.....	25
Service .....	25
Copy .....	25
Debug.....	26
File .....	26
Lineinfile .....	27
Git.....	27
Cli_command .....	28
Archive .....	28
Command .....	28

## ÍNDICE DE TABLAS

Tabla 1 - Infraestructura necesaria en el centro docente .....	18
Tabla 2 - ACNEAE. Dificultades y posibles soluciones .....	19
Tabla 3 - Impedimentos y posibles soluciones.....	20

## ÍNDICE DE FIGURAS

Ilustración 1 -Topología de red en el centro docente .....	10
Ilustración 2 - Configuración de repositorios .....	11
Ilustración 3 - Instalación servidor ansible.....	12
Ilustración 4 - Configuración de red del servidor .....	13
Ilustración 5 - Configuración de red cliente 1 .....	14
Ilustración 6- Configuración de red cliente 2 .....	14
Ilustración 7 - Despliegue con errores .....	17
Ilustración 8 Despliegue con resultado correcto .....	17

## DECLARACIÓN DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE FIN DE MÁSTER

Don ROBERTO ANGUITA MARTÍN, con DNI 09017829N, estudiante del MÁSTER EN PROFESORADO DE EDUCACIÓN SECUNDARIA OBLIGATORIA Y BACHILLERATO, FORMACIÓN PROFESIONAL Y ENSEÑANZA DE IDIOMAS de la Universidad de Alicante, realizado en el período 2020-2021.

### DECLARA QUE:

La Memoria del Trabajo Fin de Máster denominado, **Gestión de software centralizada con Ansible en un centro docente**, ha sido desarrollado respetando los derechos intelectuales de terceros, conforme las citas que constan en las páginas correspondientes y cuyas fuentes se incorporan en la bibliografía, así como cualquier otro derecho, por ejemplo, de imagen que pudiese estar sujeto a protección del copyright.

En virtud de esta declaración, afirmo que este trabajo es inédito y de mi autoría, por lo que me responsabilizo del contenido, veracidad y alcance del Trabajo Fin de Máster, y asumo las consecuencias administrativas y jurídicas que se deriven en caso de incumplimiento de esta declaración.



San Vicente del Raspeig, 25 de MAYO de 2021.

## 1. INTRODUCCIÓN Y JUSTIFICACIÓN

El tema principal del Trabajo Fin de Máster, nace a partir de mi experiencia personal como profesor y como alumno en prácticas en varios centros de Secundaria y Formación Profesional. Durante la cual he podido constatar cómo el departamento de informática consume gran cantidad de horas no lectivas llamadas, “horas TIC”, en las que realizan tareas repetitivas de actualización, instalación de software requerido o mantenimiento, tanto de las aulas informáticas, como de los equipos del centro y tabletas del centro.

Existen distintas normativas de la Conselleria de Educación que definen los equipamientos mínimos que deben existir y otras directrices al respecto hardware y software instalado en un centro docente. Los decretos 35/2018, de 23 de marzo o 16/2012, de 20 de enero, definen que todas las compras de equipos informáticos, dominios o comunicaciones están centralizadas en la DGTIC (Direcció General de Technologies de la Informació). Así mismo las (*Instrucciones técnicas, 2011*) indican el equipamiento mínimo que tendrá cada centro docente.

Con respecto a las licencias del software, será la Conselleria de Educación la que establecerá qué licencias de software deben utilizar los centros docentes tanto para uso educativo como para uso administrativo.

Con respecto a la normativa de seguridad para el correcto uso y gestión de las líneas de datos, (*Instrucción de servicio número 5/2012, 2012*), define que no está permitido conectar un equipo simultáneamente a la red de secretaría y la red de las aulas, conectando un equipo simultáneamente a ambas líneas, así como queda totalmente prohibido interconectar las redes existentes en un centro con otras redes en las que la Generalitat Valenciana no tenga su titularidad. La (*Instrucción de servicio número 5/2012, 2012*) se tendrá en cuenta en este Trabajo Fin de Máster.

Las (*Instrucciones de servicio, s. f.*) establecen que cualquier incidencia debe comunicarse al SAI y que los servicios asistenciales suministrados por el SAI en lo referente a las aplicaciones informáticas (software), son los siguientes;

- Aplicaciones de gestión de centros (ITACA, GesCen, PGA...)
- Distribución LliureX
- Correo electrónico.
- Antivirus

Por ello todo software necesario para la realización de alguna práctica o aplicación distinta a las ofrecidas por GVA debe ser instalado por un usuario del centro con privilegios y que conozca la normativa existente. Es por ello por lo que se asignan horas no lectivas para que el departamento de informática pueda realizar las labores de mantenimientos preventivos, actualizaciones, parcheo de seguridad o instalaciones en las aulas de informática u otros departamentos. En algunos centros estas labores son contratadas a empresas externas, bajo la previa autorización de la Conselleria.

El objetivo del presente Trabajo Fin de Máster es conseguir la reducción del tiempo empleado para la ejecución de tareas repetitivas mediante la utilización de estándares de gestión empleados en la empresa privada. Para ello, se centralizará la gestión de todo el software de un centro docente con un servidor Ansible, creando distintos perfiles de configuración en función de las peculiaridades distintivas que pueda tener o no cada departamento.

En el presente Trabajo Fin de Máster se elaborará a modo de ejemplo la instalación del servidor Ansible y un ejemplo de cómo se distribuye con varios clientes simultáneamente.

## **2. DESARROLLO**

### **2.1. Contextualización**

En la mayoría de los centros educativos, los profesores del departamento de informática tienen asignadas horas no lectivas, llamadas horas TIC, para la realización de tareas de administración y mantenimiento, en el instituto donde realicé mis prácticas, IES Azorín, pude constatar mayoría de las tareas que realizaba el departamento de informática en las horas TIC, como son:

- Administración de los sistemas Lliurex.
- Actualización de los sistemas Lliurex.
- Instalación de software libre o autorizado en los ordenadores del centro.
- Instalación de libros digitales en las tabletas del centro.
- Gestión de usuarios en las aulas de informática.
- Mantenimiento de la página web del centro.
- Resolución de incidentes o problemas en los ordenadores del centro.
- Atender las dudas o consultas del resto de docentes sobre el uso de las herramientas TIC del centro.

Además de estas horas, pueden realizar alguna tarea más a petición del equipo directivo, por lo que las horas TIC no son suficientes para cubrir la actual demanda.

Las horas TIC se definen para los coordinadores TIC y los coordinadores del aula de informática según queda establecido en el punto 3.7.2 de (RESOLUCIÓN de inicio de curso, 2019)

Sus funciones serán las siguientes:

- Coordinar el uso del aula o aulas del centro.
- Velar por el mantenimiento del material informático.
- Asesorar en materia informática al resto del profesorado e informar de las actividades que se lleven a cabo en el aula o aulas de informática.
- Confeccionar el inventario de máquinas y material informático y responsabilizarse que esté disponible y en óptimas condiciones de utilización.

La ley establece para el desarrollo de estas funciones, a la persona coordinadora del aula de Informática se le asignan las siguientes horas lectivas según lo que dispone el artículo 4.9 de la Orden 69/2015, de 25 de junio:

- a) Centros que tengan entre 1 y 9 grupos de Informática y TIC: 2 horas lectivas.
- b) Centros que tengan entre 10 y 17 grupos de Informática y TIC: 4 horas lectivas.
- c) Centros con 18 o más grupos de Informática y TIC: 6 horas lectivas.

Los coordinadores TIC, tienen asignadas unas horas adicionales para realizar otras tareas adicionales, dispuesto en el punto 3.73 de la (RESOLUCIÓN de inicio de curso, 2019) y en el artículo 51 del (DECRETO 252/2019, 2019), las cuales son:

- Coordinar el uso del aula o aulas de informática del centro.
- Velar por el mantenimiento del material informático.
- Ejercer la interlocución con el apoyo y la asistencia informática (SAI) así como con la administración, en los temas relativos a las TIC.
- Gestionar, dentro de la aplicación de inventario TIC proporcionada por la Administración, el hardware y el software de que dispone el centro, se responsabiliza de que esté localizado y disponible.
- Colaborar en la confección del inventario del material informático no incluido en la aplicación de inventario TIC.
- Promover y asesorar al claustro en la integración de las TIC en la tarea docente y su incorporación a la planificación didáctica y proyectos de innovación.
- Asesorar en materia informática al resto del profesorado e informar de las actividades que se lleven a cabo en el aula o aulas de informática.
- Conocer, promover y asesorar a la comunidad educativa en el uso de las aplicaciones de gestión académica, administrativa y de comunicación que la Administración ponga al alcance de los centros educativos.
- Conocer y asesorar a la comunidad educativa sobre la normativa y la ordenación reguladora del uso de las TIC en el centro.
- Colaborar en el diseño del programa anual de formación del centro asesorando al claustro, y específicamente a la figura de la coordinación de formación, para realizar el análisis de las necesidades formativas del claustro en relación con el ámbito TIC.
- Asesorar a la secretaría del centro en su función de atender y difundir los requisitos normativos para que el centro cumpla la normativa vigente en protección de datos en los procedimientos administrativos y en la manera de



compartir información mediante las tecnologías de la información y la comunicación.

- Cualquier otra que la Administración educativa determine en su ámbito de competencias.

Para el desarrollo de estas funciones, a la persona coordinadora TIC se le asignan 2 horas lectivas y 2 horas complementarias según lo que dispone el artículo 4.10 de la Orden 69/2015, de 25 de junio.

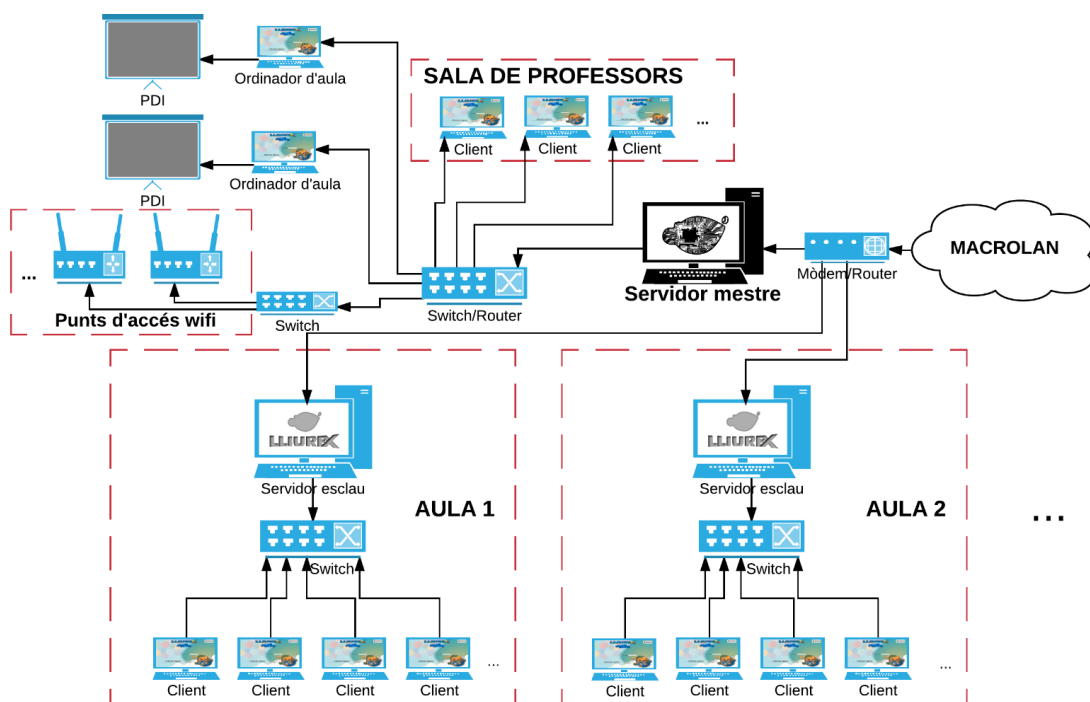
A partir de estas tareas, el presente proyecto pretende conseguir la reducción de tiempos empleados en la realización de tareas repetitivas en los ordenadores del centro. Tareas como una simple actualización de paquetes en los ordenadores del centro requiere que el docente se conecte uno por uno a cada ordenador ejecutando la tarea de actualizar los paquetes, por lo que el tiempo necesario para realizar esta tarea podría formularse de esta forma:

$$(Tiempo\ de\ conexión + Tiempo\ de\ ejecución) * número\ de\ ordenadores$$

Con este proyecto se pretende reducir el tiempo empleado en este tipo de tareas, ejecutando simultáneamente en todos los ordenadores de forma asíncrona las tareas (mantenimiento, actualización, instalación, etc....). Podríamos formular el tiempo necesario tras aplicar esta propuesta de la siguiente forma:

$$(Tiempo\ de\ desarrollo\ de\ la\ configuración\ Ansible + Tiempo\ de\ conexión + Tiempo\ de\ ejecución)$$

Basándonos en el modelo actual de la mayoría de los centros docentes, este proyecto utilizará la última distribución de Lliurex de la siguiente forma:



Il·lustració 1 -Topologia de red en el centre docente

En el presente TFM se elaborará a modo de ejemplo la instalación del servidor Ansible y cómo se distribuye y configura un servidor de páginas web y otro de bases de datos, para que los alumnos puedan realizar una práctica concreta en los ordenadores del aula.

## 2.2. ¿Qué es Ansible?

Ansible es una plataforma de software libre, con licencia GNU GPL v3(*The GNU General Public License v3.0 - GNU Project - Free Software Foundation, 2007*) que permite configurar y administrar ordenadores. Combina las funcionalidades de instalación multi-nodo, ejecuciones de tareas y administración de configuraciones. Adicionalmente, Ansible está categorizado como una herramienta de orquestación, resulta muy útil para los administradores de sistema y DevOps, porque a tenor de estas características puede ser muy útil para su uso dentro de un centro educativo, por los administradores de sistemas del centro y los profesores de informática. Esta herramienta fue programada en Python por Michael DeHanny posteriormente patrocinada por Redhat. Ansible se distribuye en Fedora, Red Hat enterprise Linux, CentOS y Scientific Linux mediante los paquetes EPEL, además está disponible para diferentes distribuciones Linux aparte de las anteriores mencionadas como es Lliurex.

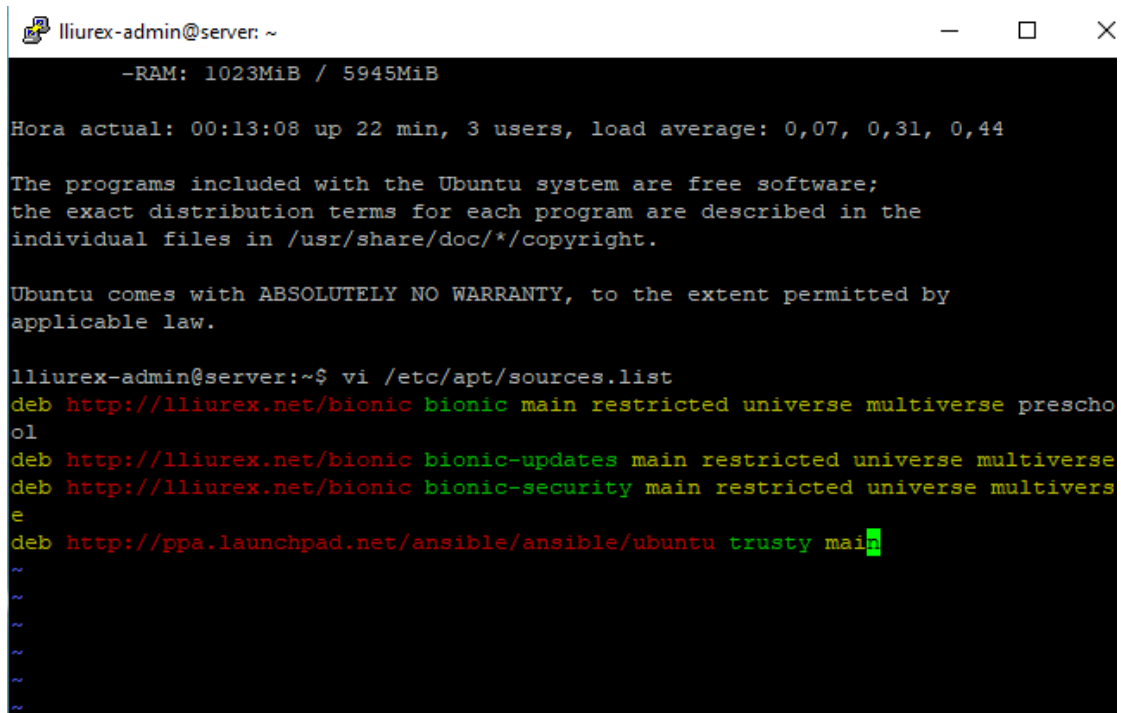
### 2.3. Instalación del servidor

Siguiendo los pasos de la guía de instalación oficial de Ansible (*Installation Guide — Ansible Documentation*, 2021)

Procedemos a instalar el servidor Ansible en una distribución Lliurex. A continuación, mostramos todos los pasos necesarios:

Paso 1 - Añadimos la siguiente línea al fichero `/etc/apt/sources.list` para añadir el repositorio de Ansible para debían:

```
deb http://ppa.launchpad.net/ansible/ansible/ubuntu trusty
main
```



```
lliurex-admin@server: ~
-RAM: 1023MiB / 5945MiB

Hora actual: 00:13:08 up 22 min, 3 users, load average: 0,07, 0,31, 0,44

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

lliurex-admin@server:~$ vi /etc/apt/sources.list
deb http://lliurex.net/bionic bionic main restricted universe multiverse prescho
ol
deb http://lliurex.net/bionic bionic-updates main restricted universe multiverse
deb http://lliurex.net/bionic bionic-security main restricted universe multivers
e
deb http://ppa.launchpad.net/ansible/ansible/ubuntu trusty main
~
~
~
~
~
~
```

Ilustración 2 - Configuración de repositorios

Paso 2 - Ejecutamos los siguientes tres comandos para actualizar el repositorio de software y proceder a instalar el paquete de Ansible.

```
$ sudo apt-key adv --keyserver keyserver.ubuntu.com --
recv-keys 93C4A3FD7BB9C367
$ sudo apt update
$ sudo apt install -y ansible
```

```

lliurex-admin@server: ~
lliurex-admin@server:~$ sudo apt-key adv --keyserver keyserver.ubuntu.com --recv
-keys 93C4A3FD7BB9C367
Executing: /tmp/apt-key-gpghome.pvw8gQlhgV/gpg.1.sh --keyserver keyserver.ubuntu
.com --recv-keys 93C4A3FD7BB9C367
gpg: clave 93C4A3FD7BB9C367: clave pública "Launchpad PPA for Ansible, Inc." imp
ortada
gpg: Cantidad total procesada: 1
gpg:      importadas: 1
lliurex-admin@server:~$ sudo apt update
Des:1 http://lliurex.net/bionic bionic InRelease [10,7 kB]
Des:2 http://lliurex.net/bionic bionic-updates InRelease [9.175 B]
Des:3 http://ppa.launchpad.net/ansible/ansible/ubuntu trusty InRelease [16,0 kB]
Des:4 http://lliurex.net/bionic bionic-security InRelease [9.155 B]
Des:5 http://lliurex.net/bionic bionic/main i386 Packages [983 kB]
Des:6 http://lliurex.net/bionic bionic/main amd64 Packages [1.963 kB]
Des:7 http://ppa.launchpad.net/ansible/ansible/ubuntu trusty/main amd64 Packages
[692 B]
Des:8 http://ppa.launchpad.net/ansible/ansible/ubuntu trusty/main i386 Packages
[692 B]
Des:9 http://ppa.launchpad.net/ansible/ansible/ubuntu trusty/main Translation-en
[492 B]
Des:10 http://lliurex.net/bionic bionic/universe amd64 Packages [724 kB]
Des:11 http://lliurex.net/bionic bionic/universe i386 Packages [723 kB]
Des:12 http://lliurex.net/bionic bionic/preschool i386 Packages [4.691 B]
Des:13 http://lliurex.net/bionic bionic/preschool amd64 Packages [4.691 B]
Des:14 http://lliurex.net/bionic bionic-updates/universe amd64 Packages [134 kB]
Des:15 http://lliurex.net/bionic bionic-updates/universe i386 Packages [133 kB]
Des:16 http://lliurex.net/bionic bionic-security/universe i386 Packages [74,9 kB
]
Des:17 http://lliurex.net/bionic bionic-security/universe amd64 Packages [74,8 k
B]
Descargados 4.865 kB en 2s (2.839 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se pueden actualizar 8 paquetes. Ejecute «apt list --upgradable» para verlos.
lliurex-admin@server:~$ sudo apt install -y ansible
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  python-httpplib2
Se instalarán los siguientes paquetes NUEVOS:
  ansible python-httpplib2
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 8 no actualizados.
Se necesita descargar 5.835 kB de archivos.
Se utilizarán 57,4 MB de espacio de disco adicional después de esta operación.
Des:1 http://ppa.launchpad.net/ansible/ansible/ubuntu trusty/main amd64 ansible all 2.9.20-lppa-trusty [5.800 kB]
Des:2 http://lliurex.net/bionic bionic-updates/main amd64 python-httpplib2 all 0.9.2+dfsg-lubuntu0.2 [34,8 kB]
Descargados 5.835 kB en 1s (7.519 kB/s)
Seleccionando el paquete python-httpplib2 previamente no seleccionado.
(Leyendo la base de datos ... 356080 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../python-httpplib2_0.9.2+dfsg-lubuntu0.2_all.deb ...
Desempaquetando python-httpplib2 (0.9.2+dfsg-lubuntu0.2) ...
Seleccionando el paquete ansible previamente no seleccionado.
Preparando para desempaquetar .../ansible_2.9.20-lppa-trusty_all.deb ...
Desempaquetando ansible (2.9.20-lppa-trusty) ...
Configurando python-httpplib2 (0.9.2+dfsg-lubuntu0.2) ...
Configurando ansible (2.9.20-lppa-trusty) ...
Procesando disparadores para man-db (2.8.3-2ubuntu0.1) ...
lliurex-admin@server:~$

```

*Ilustración 3 - Instalación servidor ansible*

También es posible instalarlo de forma multiplataforma usando el módulo PIP de Python:

```
$ python -m pip install --user ansible
```

Paso 3 - Comprobamos la configuración de red del servidor que se usará en la configuración posterior.

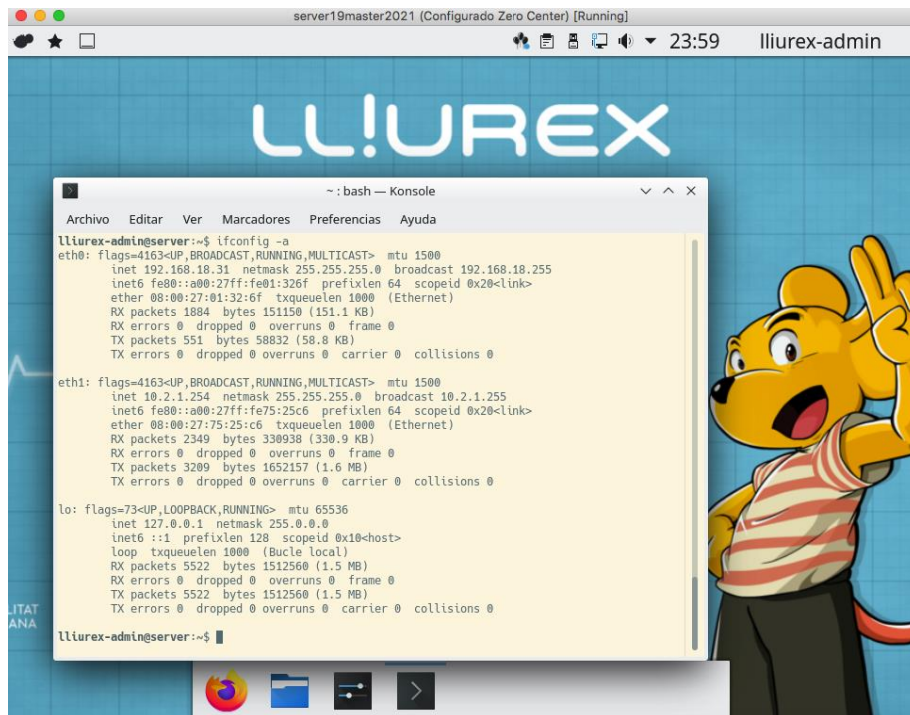


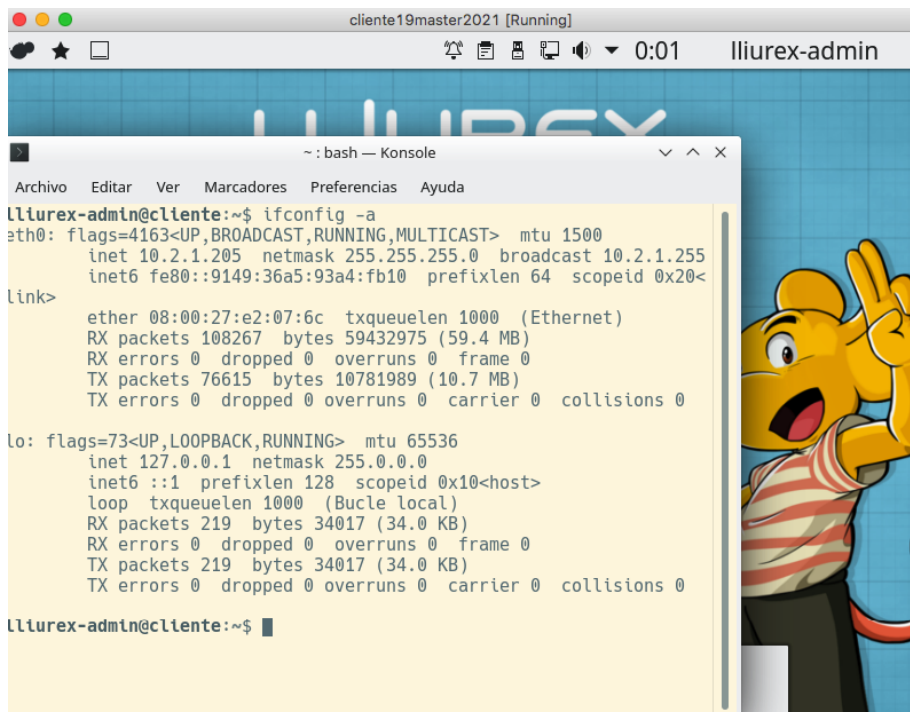
Ilustración 4 - Configuración de red del servidor

## 2.4. Instalación del cliente

Siguiendo los pasos de la guía de instalación oficial de Ansible (*Installation Guide* — *Ansible Documentation*, 2021)

Verificamos que los PCs del aula tienen SSH instalado y Python.

Comprobamos la configuración de red de uno de los clientes como ilustración que se usará en la configuración posterior.



```
lliurex-admin@cliente:~$ ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.2.1.205 netmask 255.255.255.0 broadcast 10.2.1.255
    inet6 fe80::9149:36a5:93a4:fb10 prefixlen 64 scopeid 0x20<link>
        ether 08:00:27:e2:07:6c txqueuelen 1000 (Ethernet)
        RX packets 108267 bytes 59432975 (59.4 MB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 76615 bytes 10781989 (10.7 MB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Bucle local)
    RX packets 219 bytes 34017 (34.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 219 bytes 34017 (34.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lliurex-admin@cliente:~$
```

Ilustración 5 - Configuración de red cliente 1

Otro PC del Aula cliente y usado para este ejemplo tiene esta IP



```
lliurex-admin@cliente:~$ ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.2.1.177 netmask 255.255.255.0 broadcast 10.2.1.255
    inet6 fe80::99a:ee93:cfce:bc19 prefixlen 64 scopeid 0x20<link>
        ether 08:00:27:36:c9:1c txqueuelen 1000 (Ethernet)
        RX packets 107793 bytes 59167619 (59.1 MB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 76174 bytes 10716049 (10.7 MB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Bucle local)
    RX packets 201 bytes 31860 (31.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 201 bytes 31860 (31.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Ilustración 6- Configuración de red cliente 2

## 2.5. Configuración del inventario

A continuación, siguiendo la guía de usuario (*User Guide*, 2021) procedemos a configurar en el servidor donde hemos instalado Ansible, un fichero de inventario que contenga todos los hosts que serán orquestados por Ansible. Los equipos (PCs) o también llamados nodos clientes, podrán ser agrupados en grupos o roles en función de la función que realicen. En este ejemplo, solamente se dispone de 3 PCs: el servidor y dos PCs clientes de un aula. Todos ellos harán uso de la red interna de las aulas. El equipo 1 y 2 pertenece al grupo aulas y el servidor de ansible al grupo “ansibleserver”. El fichero que hay que editar es `/etc/ansible/hosts` y en el agrupamos los nodos y por roles en función de la funcionalidad que realicen de la siguiente forma de la siguiente forma:

```
[ansibleserver]
10.2.1.254

[dbservers]
10.2.1.177
10.2.1.205

[webservers]
10.2.1.177
10.2.1.205
```

## 2.6. Configuración del playbook

Los playbook son ficheros de configuración, despliegue y orquestación en Ansible. Emplean el formato YAML. Cada playbook asocia un grupo de hosts a un conjunto de roles. Cada rol está representado por llamadas a lo que Ansible define como Tareas que a su vez hacen uso de:

- Variables de entorno.
- Plantillas
- Ficheros
- Controladores

Para el ejemplo desarrollado se han definido dos roles basados en la funcionalidad que realizarán, como son “db” para el grupo definido en el inventario de los servidores de bases de datos(dbservers) y “base-apache” que son los servidores de páginas web(webrowsers), en este caso ambos roles estarán en los ordenadores del aula de forma que cada alumno tenga su servidor de páginas web y su base de datos.

La definición del playbook sería la siguiente:

```
---
# This playbook deploys the whole application stack in
this site.

# Apply common configuration to all hosts
- hosts: all

  roles:
  - common

# Configure and deploy database servers.
- hosts: dbservers

  roles:
  - db

# Configure and deploy the web servers. Note that we
include two roles
# here, the 'base-apache' role which simply sets up
Apache, and 'web'
# which includes our example web application.

- hosts: webservers

  roles:
  - base-apache
```

La definición de las tareas, variables de entorno, plantillas y controladores realizados para este TFM como ejemplo de distribución en un aula de un servidor apache con php y una base de datos mysql en cada equipo, para que los alumnos puedan realizar unas prácticas de HTML, está publicado en mi repositorio personal de Github (*Anguita, 2021*).

Para lanzar toda la orquestación y que se realice la instalación en los equipos hay que ejecutar el siguiente comando:

```
ansible-playbook [-i <nombre del fichero de inventario en
el caso de no definirlo en /etc/ansible/hosts>] <nombre
del fichero del playbook>
```

Ejemplo empleado:

```
ansible-playbook -i host webdb.yml
```



En el caso de tener algún error, nos mostrará una salida como esta:

```
lliuirex-admin@server:/etc/ansible$ ansible-playbook -i hosts webdb.yml
PLAY [all] *****
TASK [Gathering Facts] *****
fatal: [10.2.1.254]: UNREACHABLE => ("changed": false, "msg": "Failed to connect to the host via ssh: lliurex-admin@10.2.1.254: Permission denied (publickey,password).", "unreachable": true)
fatal: [10.2.1.205]: UNREACHABLE => ("changed": false, "msg": "Failed to connect to the host via ssh: lliurex-admin@10.2.1.205: Permission denied (publickey,password).", "unreachable": true)
fatal: [10.2.1.177]: UNREACHABLE => ("changed": false, "msg": "Failed to connect to the host via ssh: lliurex-admin@10.2.1.177: Permission denied (publickey,password).", "unreachable": true)
PLAY RECAP *****
10.2.1.177      : ok=0  changed=0  unreachable=1  failed=0  skipped=0  rescued=0  ignored=0
10.2.1.205     : ok=0  changed=0  unreachable=1  failed=0  skipped=0  rescued=0  ignored=0
10.2.1.254     : ok=0  changed=0  unreachable=1  failed=0  skipped=0  rescued=0  ignored=0
lliurex-admin@server:/etc/ansible$
```

*Ilustración 7 - Despliegue con errores*

En el caso de ejecutarse todo el despliegue correctamente nos mostraría una salida resumen así:

```
TASK [base-apache : change default ports] *****
ok: [10.2.1.177]
ok: [10.2.1.205]

TASK [base-apache : install php5 and python on apache2] *****
ok: [10.2.1.205] => (item=[u'libapache2-mod-php', u'php7.2-mysql'])
ok: [10.2.1.177] => (item=[u'libapache2-mod-php', u'php7.2-mysql'])

TASK [base-apache : copy php.ini] *****
ok: [10.2.1.205]
ok: [10.2.1.177]

PLAY RECAP *****
10.2.1.177      : ok=20  changed=0  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
10.2.1.205     : ok=20  changed=0  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
10.2.1.254     : ok=1   changed=0  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
```

*Ilustración 8 Despliegue con resultado correcto*

### 3. INFRAESTRUCTURA

Los requisitos necesarios para realizar esta implantación de software son los mismos que tiene el servidor Lliurex en la actualidad puesto que no necesitamos agregar ningún hardware ni conectividad exterior adicional, únicamente variaría la apertura del puerto 22 hacia el servidor de ansible. Los datos de la infraestructura actual se han extraído de las (*Instrucciones técnicas, 2011*). Los requisitos del servidor serían, por tanto:

*Tabla 1 - Infraestructura necesaria en el centro docente*

<b>Hardware</b>	<ul style="list-style-type: none"><li>- Procesador: Pentium IV 2Ghz o superior.</li></ul> <p>Mínimo de RAM: 4 GB mínimo.</p> <ul style="list-style-type: none"><li>- Disco duro: 2 discos duros</li><li>- Tarjeta de red: 2 tarjetas de red al menos una de 1 Gb/s (recomendables las dos)</li></ul>
<b>Acceso a Internet</b>	<ul style="list-style-type: none"><li>- Banda ancha, recomendados 2 Mb/s, a ampliar según instalación.</li></ul>
<b>Red troncal</b>	No se necesita ninguna infraestructura especial y se considera que 100 Mbits/s es suficiente.
<b>Red de aula</b>	<ul style="list-style-type: none"><li>- Mínimo switch con un puerto de 1 Gb/s</li><li>- Recomendable switch con puertos 1Gb/s</li><li>- Apertura del puerto 22 (ssh) y redirección al servidor ansible de la red de aulas.</li></ul>
<b>Red secretaria</b>	<ul style="list-style-type: none"><li>- Mínimo switch con un puerto de 1 Gb/s</li><li>- Recomendable switch con puertos 1Gb/s</li><li>- Apertura del puerto 22 (ssh) y redirección al servidor ansible de la red de secretaria.</li></ul>

#### 4. POSIBLES IMPEDIMENTOS

Hay que tener en cuenta las competencias digitales que puede tener el profesorado ya que en este caso el alumnado no va a acceder al servidor de Ansible.

La distribución con Ansible es sencilla, ya que la parametrización y orquestación se realiza con ficheros en formato YAML. Además, los PCs clientes no requieren la instalación de ningún agente o software, solamente el servicio ssh arrancado. No se requiere de conocimiento de redes, solo debe tener en cuenta la apertura del puerto 22 del ssh desde los PCs clientes al servidor Ansible correspondiente. Los servidores de Ansible no son necesarios que tengan conexión a Internet, ni deben estar expuestos, por lo que la conexión se realizará siempre desde la red local del centro, por lo que no es necesario tener en cuenta conexiones wifi o de dispositivos smartphone.

En particular se destaca la siguiente dificultad:

*Tabla 2 - ACNEAE. Dificultades y posibles soluciones*

<b>FÍSICAS O SENSORIALES</b>	<p><b>Problema:</b> Docentes con deficiencias visuales.</p> <p><b>Solución:</b> se facilitará la visibilidad del terminal de Linux necesario para ejecutar los comandos de ansible, para ello se ampliará el tamaño de la fuente empleada por el sistema en los terminales y se instalará un editor de código que mejore la visualización y edición de los ficheros YAML, como por ejemplo pueden ser Visual Code o Sublime Text</p>
------------------------------	--

Se han elaborado una serie de tutoriales a seguir por los docentes que carezcan de competencias digitales en YAML (Anexo 1) o Ansible (Anexo 2).

Para aquellos que carezcan de competencias digitales en Python se recomienda seguir el manual (*Uniwebsidad, s. f.*)

## 5. CASUÍSTICAS

Durante el uso o programación de nuevas distribuciones u orquestaciones se han detectado algunos posibles problemas que puedan impedir el uso de la herramienta, así como algunas posibles soluciones a dichos problemas.

*Tabla 3 - Impedimentos y posibles soluciones*

<b>PÉRDIDA DEL PROYECTO O BORRADO ACCIDENTAL</b>	Se realizarán copias completas periódicas automatizadas e incrementales del servidor de Ansible, no obstante, es conveniente guardar los proyectos en un repositorio de código en el propio servidor Ansible o en otro servidor del centro.
<b>INCOMPATIBILIDAD DE VERSIONES</b>	Puede encontrarse comportamientos inesperados en el caso de actualizar unos clientes y otros no y cambiar la versión de Python, por lo que se recomienda refrescar el repositorio mirror de los PCs de aulas o secretaria simultáneamente y actualizar los sistemas periódicamente en todos los equipos.

## **6. CONCLUSIONES / IMPLICACIONES**

La situación actual en la mayoría de los centros es similar, sobre todo, en los centros donde no se imparte módulos de informática de formación profesional, ya que el profesorado de informática es mucho más reducido.

En la mayoría de los centros docentes se tiende a tener la idea errónea que el profesor de informática es el técnico que repara cualquier problema que exista con los ordenadores y tabletas del centro. Esta idea errónea de confundir al profesor de informática con el SAT de Conselleria produce una gran carga de trabajo sobre el departamento de informática que, en mayor o menor medida, al menos en los centros en los que he podido asistir, hace que los profesores del departamento eviten relacionarse en la sala de profesores con otros docentes para que no les reclamen su atención con sus problemas informáticos.

En los últimos años en el aula, se ha multiplicado de forma exponencial el uso de software, mostrando en muchos casos las carencias actuales del profesorado en competencias digitales, como son el manejo de Aules, Lliurex u otros recursos de GVA. Debido a la pandemia, se ha hecho más necesario la utilización de herramientas interactivas, o de comunicación para poder impartir las clases, esto ha producido un incremento de las consultas realizadas al departamento de informática para suplir las deficiencias digitales del profesorado. Este punto, entra dentro de las funciones que tiene asignadas en primera persona el coordinador TIC, tiene como una de las funciones optimizar el uso de TIC en el centro y dinamización curricular en artículo 3.73 (RESOLUCIÓN de inicio de curso, 2019)

Con el sistema centralizado de distribución de software y configuraciones propuesto se pretende reducir el número de horas dedicadas por el departamento de informática para la realización de tareas repetitivas. Con este ahorro de tiempo obtenido se podrá atender en la mayoría de los casos la demanda de incidencias, parcheo o instalaciones que de otra forma no podrían atenderse en las horas asignadas para estos trabajos. Con ello permitirá por una parte al departamento de informática poder integrarse más con el resto del profesorado y poder ayudar al coordinador TIC a dinamizar el uso de las TIC en el currículum.

## 7. PROPUESTA DE MEJORA PARA LA IMPLEMENTACIÓN

Como propuesta de mejora a la implementación indicada, recomiendo las siguientes mejoras:

1. Implementación de servidor GIT en la red de Aulas y otro en la red de secretaria, donde y creación de varios repositorios de Ansible donde se almacenarán las distintas configuraciones, tareas, variables y playbooks aplicadas en el centro. Sirva de ejemplo mi repositorio personal de Ansible en de Github (*Anguita, 2021*).
2. Implementación de una Wiki o gestor documental donde se expliquen y documenten los valores aplicados en los ficheros YAML.

## 8. BIBLIOGRAFÍA

- All modules — Ansible Documentation. (2021, 30 abril). Ansible Documentation.  
[https://docs.ansible.com/ansible/2.8/modules/list\\_of\\_all\\_modules.html](https://docs.ansible.com/ansible/2.8/modules/list_of_all_modules.html)
- Anguita, R. (2021, 14 mayo). Repositorio personal de github ansible-apache-mysql. GitHub. <https://github.com/ranguitamartin/ansible-apache-mysql>
- Installation guide — Ansible documentation. (2021, 23 abril). Ansible.  
[https://docs.ansible.com/ansible/latest/installation\\_guide/](https://docs.ansible.com/ansible/latest/installation_guide/)
- DECRETO 252/2019. (2019, 29 noviembre). Conselleria de Educación, Cultura y Deporte.  
[https://ceice.gva.es/documents/162640785/162670691/2019\\_11616.pdf](https://ceice.gva.es/documents/162640785/162670691/2019_11616.pdf)
- Instrucción de servicio número 5/2012. (2012, 30 julio). GVA.  
[https://ceice.gva.es/documents/161862998/163024947/Instrucción+servicio+5\\_2012+C.pdf](https://ceice.gva.es/documents/161862998/163024947/Instrucción+servicio+5_2012+C.pdf)
- Instrucciones de servicio. (s. f.). Normativa - Generalitat Valenciana.  
<https://ceice.gva.es/es/web/innovacion-tecnologica/normativa>
- Instrucciones técnicas. (2011, enero). GVA.  
[https://ceice.gva.es/documents/161863110/163204930/instruccion2011\\_val.pdf](https://ceice.gva.es/documents/161863110/163204930/instruccion2011_val.pdf)
- RESOLUCIÓN de inicio de curso. (2019, 5 julio). Diari oficial de la Generalitat Valenciana. [https://dogv.gva.es/datos/2019/07/18/pdf/2019\\_7221.pdf](https://dogv.gva.es/datos/2019/07/18/pdf/2019_7221.pdf)

- The GNU General Public License v3.0 - GNU Project - Free Software Foundation. (2007, 29 junio). GNU. <https://www.gnu.org/licenses/gpl-3.0.html>
- Uniwebsidad. (s. f.). Python para principiantes. Python para principiantes. <https://uniwebsidad.com/libros/python>
- User Guide — Ansible Documentation. (2021, 23 abril). Ansible. [https://docs.ansible.com/ansible/latest/user\\_guide/](https://docs.ansible.com/ansible/latest/user_guide/)

## 9. ANEXOS

### Anexo 1: Ficheros YAML

YAML es un formato de serialización de datos. Por lo que los ficheros YAML nos permiten guardar información de forma estructurada y serializada.

La Sintaxis básica de YAML es la siguiente:

Cada YAML comienza con --- que denota el inicio de un archivo YAML.

A continuación, se escriben todos los datos de forma clave: valor, al contrario de lo que ocurre en otros formatos de serialización como son JSON o XML los espacios se tienen en cuenta en YAML, este es un dato importante a tener en cuenta.

La sintaxis sugerida para los archivos YAML es utilizar 2 espacios para la sangría, pero YAML seguirá cualquier sistema de sangría que utilice el archivo individual.

Un ejemplo por tanto de este formato sería el siguiente:

```
---
Nombre: Roberto
Edad: 43
Teléfono: 655120301
```

YAML admite diferentes tipos de datos, caracteres, cadenas, números enteros, valores flotantes y colecciones como matrices y listas que se construyen a partir de tipos de datos básicos.

En YAML se permite la escritura de cadenas de caracteres en una o varias líneas, empleando el símbolo > en el campo del valor o múltiples líneas usando el símbolo |, muestra de ello es el siguiente ejemplo:

```
---
Refrán: >
  El perro de San Roque no tiene rabo
  Porque Ramon Rodríguez se lo ha cortado.
```

Las listas son uno de los tipos más usados en YAML, ya que nos permite poner múltiples valores asociados a una clave. Los valores empleados en las listas pueden ser cualquiera de los admitidos por YAML.

Las listas requieren que los valores estén anidados, con el mismo número de espacios, o que emplee el símbolo –

Un ejemplo de una lista con el símbolo – es el siguiente:

```
---
- Pera
- Manzana
- Coco
```

Un ejemplo del uso de una lista anidada

```
---
Automóviles:
  Coches:
    Hyundai
    Volkswagen
    Ford
```

YAML también proporciona & y \* símbolos como anclajes y referencias al ancla para evitar duplicaciones y conseguir por tanto que el archivo YAML sea más pequeño.

Un ejemplo del uso de estos símbolos es el siguiente:

```
---
detalle: &detalle
  Nombre: "Roberto"
  Edad: 43
Profesión: Informático
<<: * detalle
```

Este fichero traducido quedaría de la siguiente forma:

```
---
Profesión: Informático
Nombre: "Roberto"
Edad: 43
```

## Anexo 2: Módulos de Ansible

A continuación, voy a detallar algunos de los principales módulos de Ansible que más se utilizan, la información ha sido extraída de la web oficial de Ansible (*All Modules* —



*Ansible Documentation, 2021*), donde podemos encontrar la totalidad de los módulos existentes en Ansible.

## Gestor de paquetes

Existe un módulo para los gestores de paquetes más populares, como YUM, APT o DNF, que nos permite instalar, actualizar, degradar, eliminar y enumerar cualquier paquete en el sistema. Los nombres de los módulos relevantes son fáciles de adivinar, ya que se nombran por el <nombre del gestor>\_module por ejemplo, dnf\_module, yum\_module o apt\_module. Un ejemplo de instalación con este módulo sería:

```
- name: Instalar la última versión de Apache y MariaDB
  apt:
    name:
      - httpd
      - mariadb-server
    state: latest
```

## Service

El **módulo service** nos permite iniciar, detener o consultar el estado de los servicios instalados. Un ejemplo de su uso sería el siguiente:

```
---
- name: Reiniciar el servicio de red en la interfaz eth0
  service:
    name: network
    state: restarted
    args: eth0
```

## Copy

Este **módulo copy**, copia un archivo desde el servidor ansible o máquina remota (PC cliente) a una ubicación en la máquina remota. Un ejemplo de su uso es el siguiente:

```
---
- name: Copia el fichero "ntp.conf" en la ubicación,
  copia de seguridad del original si difiere de la versión
  copiada
  copy:
    src: /mine/ntp.conf
    dest: /etc/ntp.conf
    owner: root
    group: root
    mode: '0644'
    backup: yes
```

## Debug

El **módulo debug** imprime por pantalla las distintas declaraciones realizadas en los ficheros durante la ejecución del playbook y puede ser útil para depurar variables o expresiones sin tener que detener el playbook. Un ejemplo de su uso es el siguiente:

```
name: Muestra las variables / acciones conocidas para un
host
  debug:
    var: hostvars[inventory_hostname]
    verbosity: 2
```

Esto registra el contenido de la salida del módulo y lo muestra solo cuando especifica verbosidad como 2. Un ejemplo de su invocación con verbosity 2 es el siguiente:

```
ansible-playbook demo.yaml -vv
```

## File

El **módulo file** maneja los ficheros y sus propiedades, estableciendo atributos de archivos, enlaces simbólicos o directorios y también eliminando archivos, enlaces o directorios. Un ejemplo de su uso sería el siguiente:

```
---
- name: Cambia el propietario, grupo y permisos
  file:
    path: /etc/master.conf
    owner: roberto
    group: roberto
    mode: '0644'
```

## Lineinfile

El **módulo lineinfile** realiza el manejo de líneas de un fichero de texto.

- Emplea expresiones regulares para realizar la selección y sustitución de una línea particular en un archivo comenzando por el final hacia atrás.
- Es principalmente útil cuando desea cambiar solo una línea en un archivo, ya que comandos como “sed” realizan la misma función reemplazando la totalidad de las apariciones en un fichero.

Un ejemplo de su uso sería el siguiente:

```
- name: Indica el modo enforcing para SELINUX
  lineinfile:
    path: /etc/selinux/config
    regexp: '^SELINUX='
    line: SELINUX=enforcing
```

## Git

El **módulo git** gestiona las versiones de git en los repositorios para implementar archivos o software. Un ejemplo de su uso sería el siguiente:

```
# Crear el repositorio a partir de un fichero dado y
posterior clonado.
- git:
    repo: https://github.com/ranguitamartin/ansible-
    apache-mysql.git
    dest: /src/ansible-apache-mysql
    archive: /tmp/ansible-apache-mysql.zip
```

## Cli\_command

El **módulo cli\_command**, está disponible desde la versión de Ansible 2.7, proporciona una forma independiente de enviar configuraciones basadas en texto a dispositivos de red. El módulo manda comandos a los dispositivos de red y devuelve la lectura realizada en esos dispositivos. Emplea para ello el complemento de conexión network\_cli. Un ejemplo de su uso es el siguiente:

```
---
# Corre el comando para ver la versión y obtener la
salida en formato Json
- name: ejecutar comando con formateo a json
  cli_command:
    command: show version | json
```

## Archive

El **módulo archive** crea un archivo comprimido de uno o más archivos. Por defecto, supone que la fuente de compresión existe en el destino. Un ejemplo de su uso es el siguiente:

```
---
- name: Crear un fichero comprimido bz2, en la ruta /src
  archive:
    path:
      - /src/carpeta1
      - /src/carpeta2
    dest: /src/archivo.tar.bz2
    format: bz2
```

## Command

El **módulo command** es uno de los módulos más básicos pero útil, procesa y ejecuta el nombre del comando seguido de una lista de argumentos delimitados por espacios en el nodo remoto. Un ejemplo de su uso es el siguiente:

```
---
- name: Cambia el directorio de trabajo a repositorio/ y
  ejecuta el comando db_owner si /database no existe.
  command: /usr/bin/make_database.sh db_user db_name
  become: yes
  become_user: db_owner
  args:
    chdir: repositorio/
    creates: /database
```

En el uso de este módulo podemos ver el empleo de “become”, igual que se empleó en la configuración de las tareas de ejemplo de este TFM, con “become” indicamos la necesidad de ejecutar como “root” un determinado comando. En este ejemplo dicho comando sería */usr/bin/make\_database*.